(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 354 585 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89114907.2

(51) Int. Cl.⁴: G06F 9/38

(22) Date of filing: 11.08.89

(30) Priority: 11.08.88 JP 198789/88

(43) Date of publication of application:
14.02.90 Bulletin 90/07

(84) Designated Contracting States:
DE FR GB

(71) Applicant: KABUSHIKI KAISHA TOSHIBA
72, Horikawa-cho Saiwai-ku
Kawasaki-shi Kanagawa-ken 210(JP)

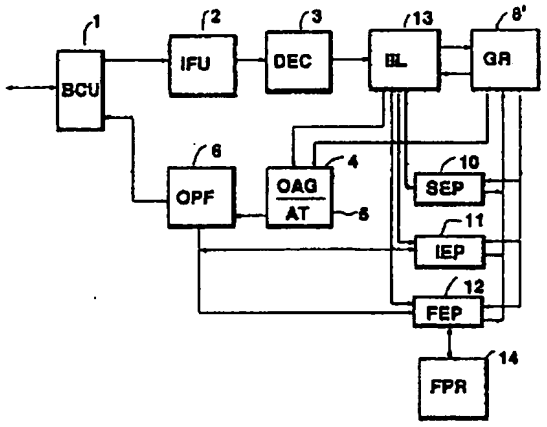
(72) Inventor: Okamoto, Kosei
6243 Hoya
Kunitachi-shi(JP)

(74) Representative: Lehn, Werner, Dipl.-Ing. et al
Hoffmann, Eitle & Partner Patentanwälte
Arabellastrasse 4
D-8000 München 81(DE)

(54) Instruction pipeline microprocessor.

(57) An instruction pipeline type microprocessor comprises an operation execution section consisting of a first operation execution section (10) for executing instructions having no memory operand, a second operation execution section (11) for executing instructions having memory operand respectively, and a third operation execution section (12) for executing floating point instructions, and a general purpose register (8') consisting of a second register group for storing data processed as a result executed by the operation execution section (10) in the form of flow of programs and a first register group for storing processed data as the result of a lock-forward execution of instructions which can be processed by the operation execution section(10) and compare and decision means for determining whether or not succeeding instructions have been executed by jumping preceding instructions is made and for sending processed data held in the first register group to the second register group so as to be exchanged along a program flow in accordance with the result of the determination.

FIG.4



EP 0 354 585 A2

INSTRUCTION PIPELINE MICROPROCESSOR

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an instruction parallel execution type microprocessor, more particularly to an instruction pipeline type microprocessor capable of rapidly executing instructions.

Description of the Prior Art

In the instruction pipeline type microprocessor according to the prior art, a particular register or registers are used in order to perform an operand address calculation in such as an ADD instruction. However, as the content of the register is modified by a previous instruction such as a TRANSFER instruction, the ADD instruction can not be shifted to an operand address calculation stage before a write stage to a general purpose register is terminated, thus delaying the processing of the instruction.

Namely, Fig. 2 shows an outline construction of the instruction pipeline type microporcessor according to the prior art.

In Fig. 2, reference numeral 1 indicates a bus control section (BCU) for connecting a microprocessor P to an external circuit, numeral 2 indicates an instruction fetch section (IFU), 3 indicates a decoder (DEC) for decoding instructions, 4 an operand address calculation section (OAG), 5 an address translation section (AT) for converting a logical address into a physical address, 6 an operand fetch section (OPF) for fetching an operand, 7 an operation execution section (EXU) for executing instructions, 8 general purpose register group (GR) consisting of a plurality of registers R_1, R_2, R_3, R_4, ... (not indicated).

When machine instructions shown in Fig. 1 are executed by the microprocessor P shown in Fig. 2 for instance, the timing of the instruction pipeline processing becomes the one shown in Fig. 3. As shown in Figs. 1 and 3, supposing that a_1 is an instruction which transfers the content of an address A to the register R_1 (not indicated) in GR_8, a_2 is an instruction which transfers data in the register R_3 in GR_8 to the register R_2, a_3 is an ADD instruction which transfers the content of an address B which is modified by the register R_2 to the register R_4, and a_4 is an instruction which transfers data in the register R_2 to a memory indicated by address C, the content of the register R_2 is used by the instruction a_3 in order to perform an operand address calculation. However, the content of the register R_2 is modified by the instruction a_2. As a result, the instruction a_3 can not move to OAG_4 until the transfer of the instruction a_2 to GR_8 is terminated, thus delaying the processing of the instruction a_3.

Since the operation execution section and the general purpose register group in the microprocessor according to the prior art are not duplicated, succeeding instructions can not be executed only after the instructions for updating the general purpose register group GR_8 have been completely executed.

Accordingly, the merits of the pipeline system can not be demonstrated due to stagnation of the processing flow in the instruction pipeline type microprocessors according to the prior art.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide an instruction pipeline type microprocessor in which succeeding instructions can be in a look-forward manner executed without waiting for the result of the execution of preceding instructions.

For the purpose of achieving the above object, the feature of the present invention resides in an instruction pipeline type microprocessor which comprises an operation execution section consisting of a first operation execution section for executing instructions having no memory operand, a second operation execution section for executing instructions having memory operand respectively, and a third operation execution section for executing floating point instructions, and a general purpose register consisting of a second register group for storing data processed as a result executed by the operation execution section in the form of flow of programs and a first register group for storing processed data as the result of a look-forward execution of instructions which can be processed by the operation execution section and compare and decision means for determining whether or not succeeding instructions have been executed by jumping preceding instructions is made and for sending processed data held in the first register group to the second register group so as to be exchanged along a program flow in accordance with the result of the determination.

Namely, the operation execution section is duplicated while the construction of the general pur-

pose register group is duplicated by the second register group for storing processed data along the flow of the programs and a first register group for storing the processed data as a result which has been a look-forward manner executed about processable instructions, whereby succeeding instructions are a look-forward manner executed without waiting for the execution result of the preceding instructions by updating data in the general purpose register group in order that the processed data in the first register group can be formed along the program flow, thus increasing the performance of the microprocessor.

These and other objects, features and advantages of the present invention will be more apparent from the following description of a preferred embodiment, taken in conjunction with the accompanying drawings.

FIRST DESCRIPTION OF THE DRAWINGS

Fig. 1 is one example of a program for clarifying problems which the microprocessors according to the prior art have,

Fig. 2 is the construction of the microprocessor of the instruction pipeline type according to the prior art,

Fig. 3 illustrates a flow of instruction processing in the microprocessor shown in Fig. 2,

Fig. 4 is a basic construction of the instruction pipeline type microprocessor according to the present invention,

Fig. 5 illustrates a flow of instruction processing in each section constituting the microprocessor shown in Fig. 4,

Fig. 6 is a detailed construction of the operation execution section and the general purpose register group shown in Fig. 4,

Fig. 7 illustrates a format of each instruction register shown in Fig. 6, and

Fig. 8 illustrates an input format of the register group FGR for storing the result of the instructions in the general purpose registers, processed in a look-forward manner.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 4 shows a principle construction of the instruction pipeline type microprocessor according to the present invention.

In the figure, same constructing elements as those shown in Fig. 2 are indicated by the same reference numerals. The instruction pipeline type microprocessor comprises an operation execution section consisting of a first operation execution (SEP) 10 for executing instructions having no memory operand respectively, a second operation execution section (IEP) 11 for executing instructions having memory operand respectively, and a third operation execution section (FEP) 12 for executing floating point instructions. In addition, reference numeral 13 indicates an instruction sending section for sending decoded instructions to each operation section and numeral 14 indicates a floating point register.

In the microprocessor according to the present invention shown in Fig. 4, the pipeline instructions shown in Fig. 4 are executed in a manner shown in Fig. 5. Namely, each instruction is fetched and processed for every clock; instruction a_2 having a memory operand is processed in the order from the instruction fetch section 2 → the decoder 3 → the instruction issue section 13 → the operand address section 4 → the address translation section 5 → the operand fetch section 6 to the second operation execution section 11.

Instruction a_2 having no memory operand but only a register operand is processed in the order from the instruction fetch section 2 → the decoder 3 → the instruction issue section 13 to the first operation execution section 10.

As shown in Fig. 5, since the executions of the instructions a_2 and a_1 are terminated at the fifth and sixth clocks respectively, it is necessary to avoid discrepancy between the data in the general purpose register group 8′ and the program flow. To this end, the general purpose register group 8′ comprises register group for holding resulting data program-processed which will be described later and register group for temporarily holding the execution result of instruction in the look-forward manner, i.e., the two register groups are duplicated.

In Fig. 6, the first operation execution section 10 comprises a first operator 20 and a first instruction register 21, the second operation execution section 11 comprises a second operator 22 and a second instruction register 23, and the third operation execution section 12 comprises a third operator 24 and a third instruction register 25. Each of the instruction registers 21, 23, and 25 has an instruction format shown in Fig. 7.

That is to say, in Fig. 7, OP indicates an operational function designation field of the associated operators, SR indicates a source register designation field, DR a destination register field, SA/D a source operand address or immediate data holding field, DA a destination operand address holding field, and PC an address holding field of an instruction in execution.

Returning to Fig. 6, the general purpose register group 8′ is divided into a register group CGR_i - (i = 1,2, 3, ... n) for storing data as a result processed in accordance with a program and a register group FGR_i (i = 1, 2, 3, ... n). Moreover, in Fig.

6, reference numeral 30 indicates an instruction address compare section, which is included in the general purpose register 8'.

In Fig. 8, there is shown a detailed format of the register group FGR which is consisted of three tag portions F, I, S each consisting of three bits and a data holding portion FGRi for holding each data.

Now, turning back to Fig. 6, each PC field of the instruction registers 21, 23, 25 is applied to the instruction address compare section 30 in order to determine whether or not a succeeding instruction is being executed by jumping a preceding instruction, for the comparisons of each PC field.

From the result of the comparison, when the PC field of the first instruction register 21 is not minimum "1" output signal is produced from the output C_1 of the compare section and when the PC field of the second instruction register 23 is not minimum, "1" output signal is produced from the output C_2 thereof, while when the DC field of the instruction register 25 is not minimum, "1" output signal is produced from the output C_3 thereof.

Furthermore, when the address PC of the instruction register in the operator which has executed an instruction is not minimum, "1" output signal (FGR WRITE signal) is produced from the output C_4 of the address compare section 30, while when the address PC of the instruction register is minimum, "1" output signal (CGR WRITE signal) is produced from the output C_5 of the compare section 30.

When the operation in each operator is terminated, the tag S in the register FGR is set at "1" when $C_4 = 1$, $C_1 = 1$, and the I tag is set at "1" when $C_2 = 1$, while the F tag is set at "1" when $C_3 = 1$.

The operation result from each of the operators 20, 22, and 24 are stored in the data holding section FGRi designated by the DR field of each of the instruction registers 21, 23, 25.

When the operation of each of the operators 20, 22, and 24 is terminated, the result of each operation is stored in the data holding section FGR_i and CGR_i, designated by the DR field of each instruction register subject to $C_4 = 0$, $C_5 = 1$. The data in another FGR_i in which any one of the tags F, I, and S has been set up, is designated in the corresponding CGR_i.

Accordingly, as shown in Fig. 5, in operation, the PC field of the first instruction register 21 shown in Fig. 3, contains the address of the instruction a_2 at the fifth clock at which time the instruction a_2 is to be executed and the PC field of the second instruction register 23 contains the instruction a_1. However, since an instruction has not been sent to the PC field of the third instruction register 25 from the instruction sending section 13, its PC field contains no address. As a result, no instruction is executed in the third operation execution section 12.

On the other hand, since the first instruction register 21 contains the instruction a_2 and the address PC field thereof is larger than that of the instruction a_1, $C_1 = 1$ is established and the result of the operation is written into FGR_2 while the tag S is set at "1".

The instruction a_1 is executed in the second operator 11 in the sixth clock. In this case, however, as the instruction register 21 in the first operator 10 contains no instruction because the instruction a_2 has already been executed therein, no action is performed. Accordingly, since $C_4 = 0$, $C_5 = 1$, the result of the operation of the instruction a_1 is stored in the CGR_1 and FGR_1, while the data in FGR_2 in which the tag S_1 has been set at "1" is now transferred to CGR_2 and is stored for a later use as data processed along the program flow.

As will be appreciated from the above description, the data necessary for the calculation of the effective address of the next instruction a_3 can be picked up from the FGR_2 (which corresponds to the register R_2) at the termination of the fifth clock. As a result, the instruction a_3 can be moved to the effective address calculation section 4 immediately.

Accordingly, unlike the microprocessors of this type according to the prior art, in the instruction pipeline type microprocessor according to the present invention, a sequential program processing in such that the instruction a_3 is processed only after the instructions a_1 and a_2 have been executed, is not required in the present invention.

As has been described in the foregoing, in the instruction pipeline type microprocessor according to the present invention, the operation execution section is duplicated so as to execute a plurality of instructions simultaneously, while the general purpose register group is also divided into a second register group for storing data processed along the program flow and a first register group for storing data as a result of having in look-forward manner executed a succeeding instruction.

In the instruction pipeline type microprocessor according to the present invention, there is also provided a compare and decision means whereby a decision is made whether or not the succeeding instruction is being executed by jumping the preceding instructions and the processed data which have been held in the first register group are sent to the second register group so as to replace them in the form along the program flow in accordance with the result of the decision.

Since succeeding instructions can be in look-forward manner executed without waiting for the result of the execution of the preceding instructions, the performance of the microprocessor of the

instruction pipeline type according to the present invention can be strikingly improved.

Various modification will become possible for those skilled in the art after receiving the teachings of the present disclosure without departing from the scope thereof.

Claims

1. An instruction pipeline type microprocessor, comprising:

(a) a plurality of operation execution sections for executing a plurality of decoded different kinds of instructions simultaneously in accordance with the different kinds of instructions;

(b) a first storing means for storing processed data as a result of having in look-forward manner executed instructions capable of processing by said plurality of operation execution sections;

(c) a second storing means for storing the processed data as a result of one executed by said operation execution sections in the form along a predetermined program flow; and

(d) a compare and decision means for determining whether or not succeeding instructions are executed by jumping preceding instructions among instructions successively issued and decoded along the predetermined program flow and for sending the processed data stored in said first storing means to said second storing means so as to be replaced in the form along the predetermined program flow, whereby processable succeeding instructions can be executed in a look-forward manner.

2. The microprocessor as claimed in claim 1, wherein said plurality of operation execution sections include a first operation execution section for executing instructions having no memory operand respectively, a second operation execution section for executing instructions having a memory operand respectively, and a third operation execution for executing floating point instructions.

3. The microporcessor as claimed in claim 1, wherein each of said plurality of operation execution sections has an instruction register having at least an instruction address holding field indicative of an address of an instruction decoded on a program and a destination register designation field for designating each of memory locations for the processed data, and said compare and decision means comprising a comparator for comparing each address from the address holding field of said each instruction register and for producing the result of the comparisons, and a plurality tag bits provided at said first storing means, so as to be selected by the resulting signals from the comparator and the designation of said destination register designation field, whereby the processed data in the first storing means is sent to the second storing means so as to be replaced in the form along the predetermined program flow.

4. The microprocessor as claimed in claim 2, wherein said first storing means is comprised of register group for successively storing the result of the operations about instructions executed, and said second storing means in comprised of register group for storing the results of the operations along the predetermined program flow.

FIG.1

```
a1 ..... MOV    adrs A , R1      : adrs A ──→ R1
a2 ..... MOV    R3 , R2          :      R3 ──→ R2
a3 ..... ADD    adrs B (R2) , R4 : adrs B (R2)+R4 ──→ R4
a4 ..... MOV    R2 , adrs C      :      R2 ──→ adrs C
```

FIG.2

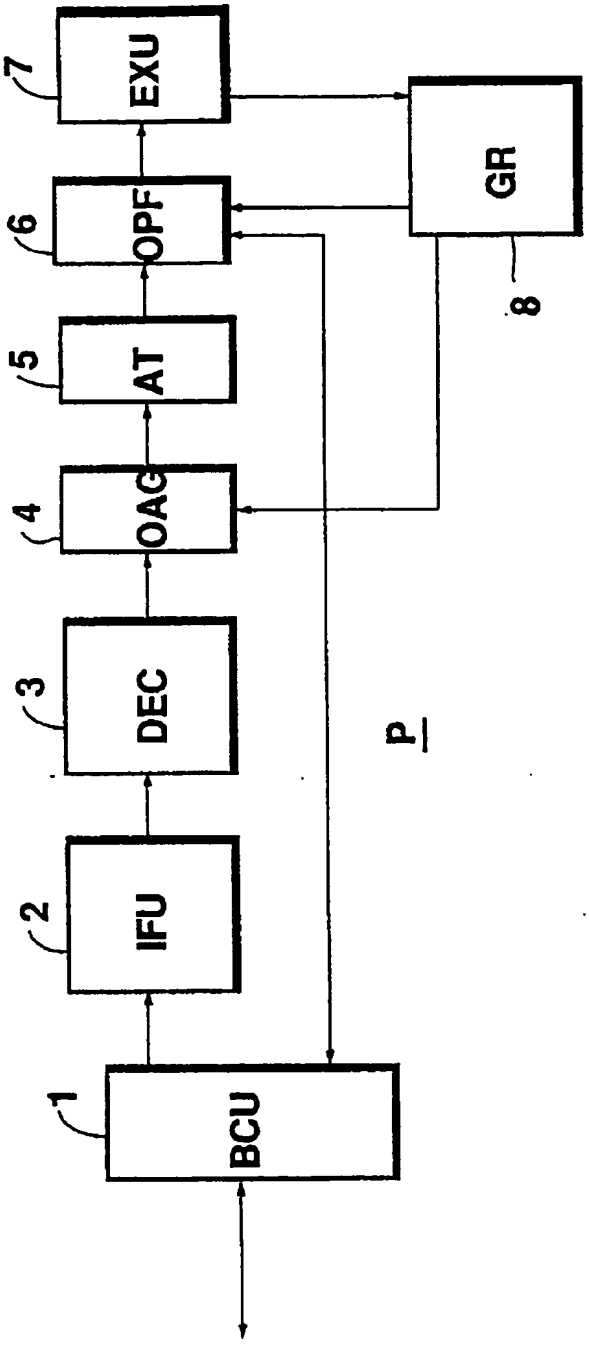


FIG.3

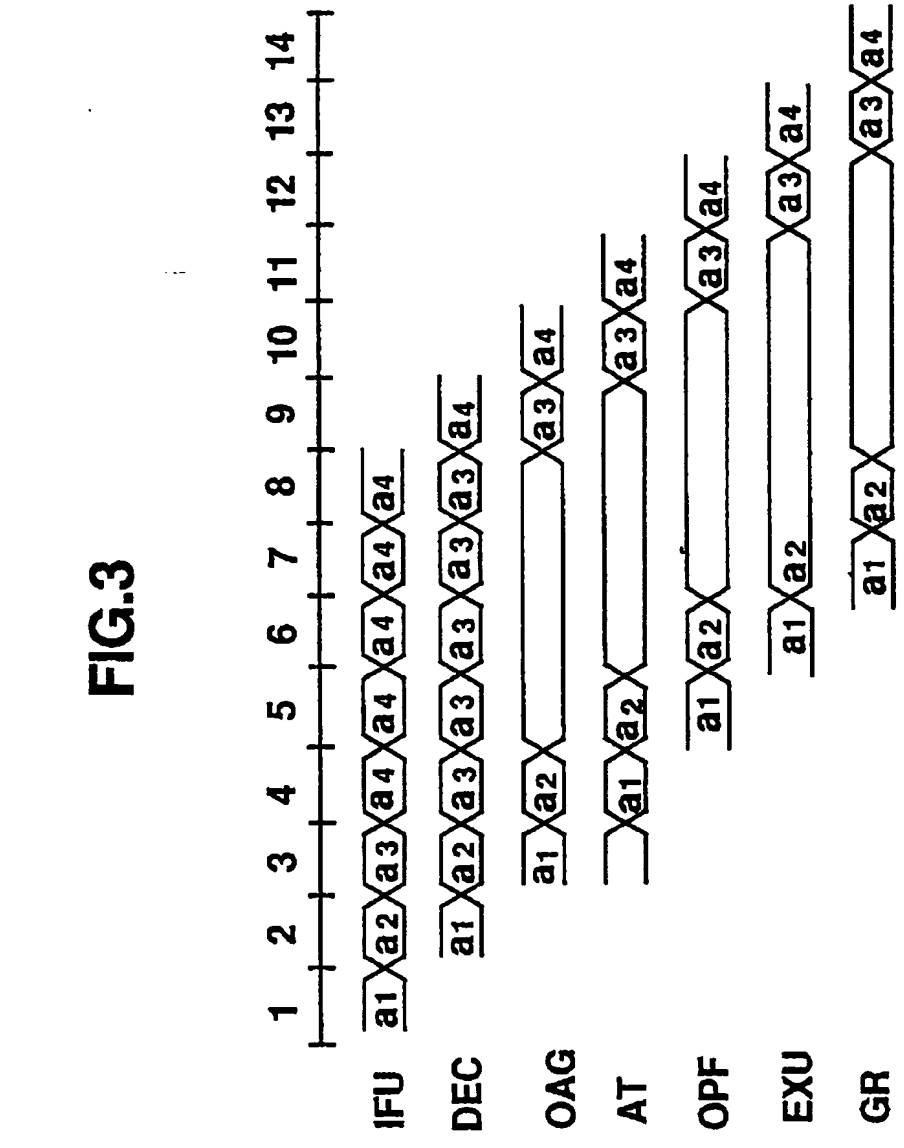


FIG.4

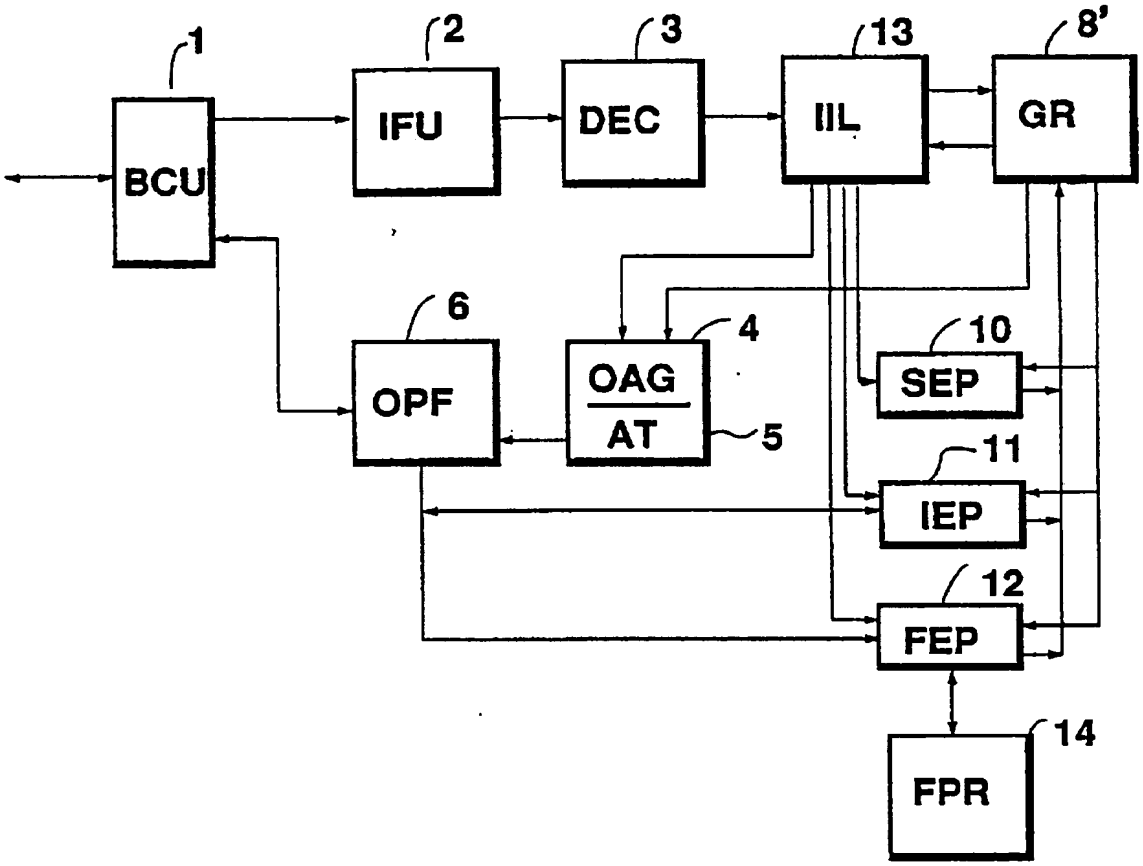
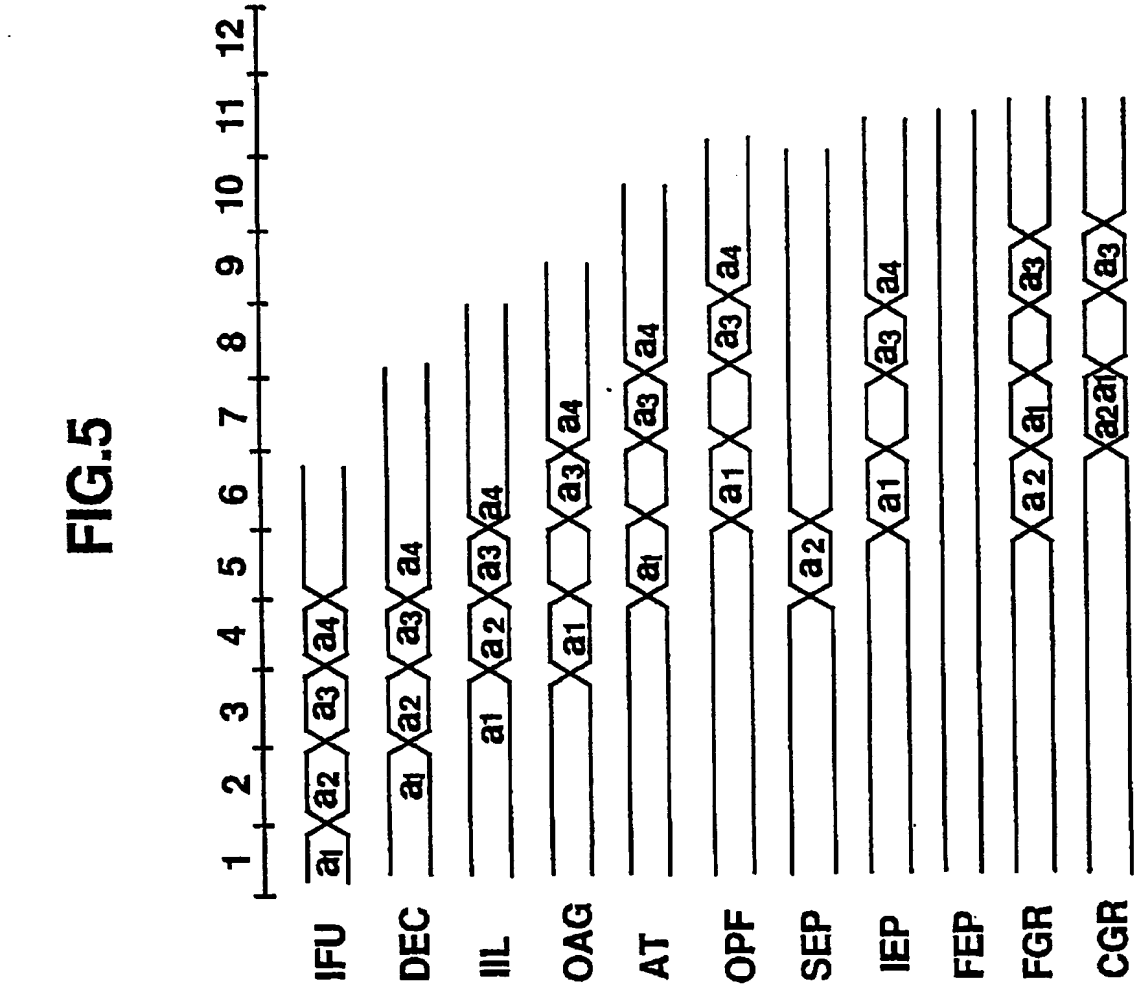

1
2
3
13
8'
BCU
IFU
DEC
IIL
GR
6
4
10
OPF
OAG
AT
5
SEP
11
IEP
12
FEP
14
FPR

FIG.5



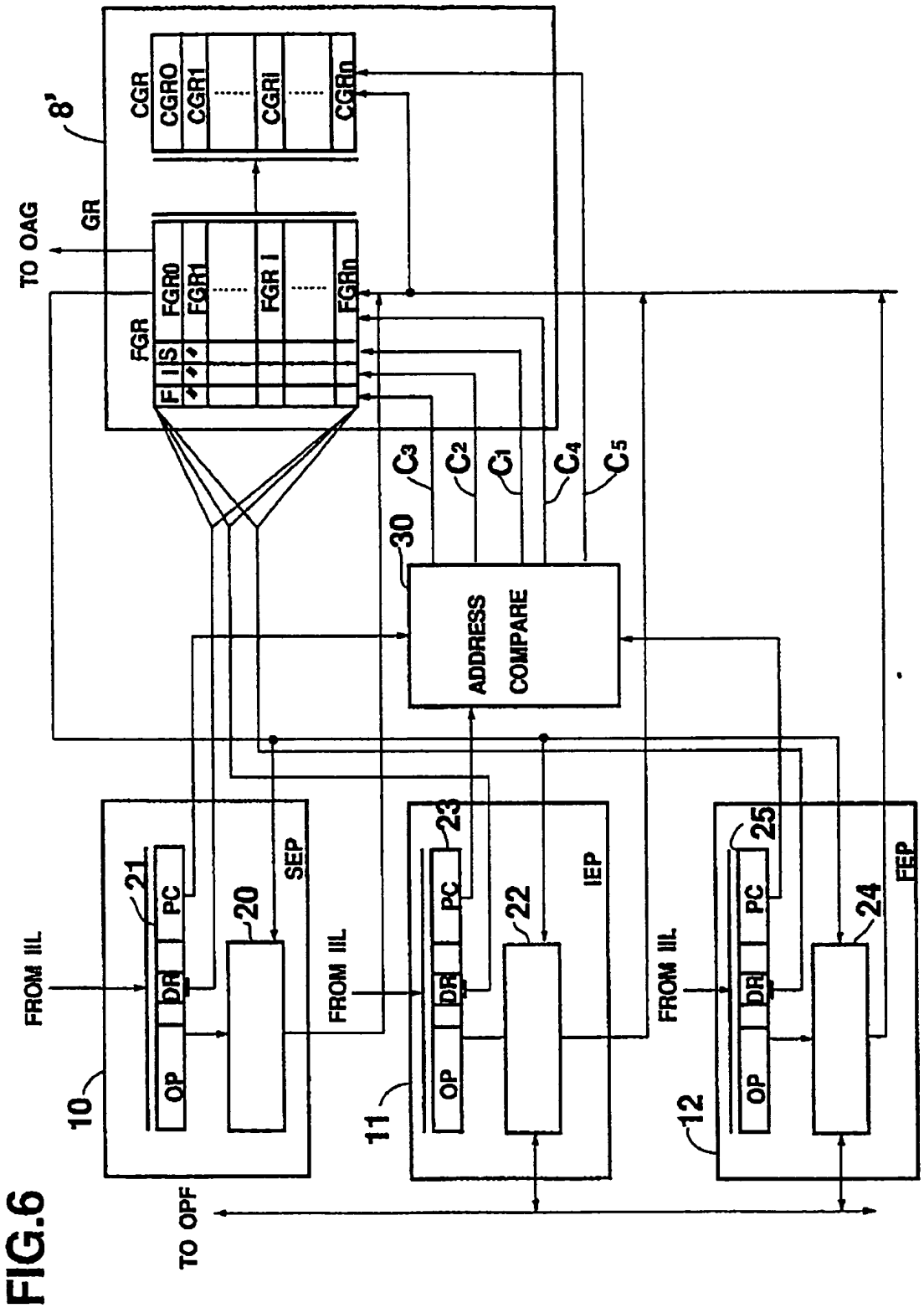

8'
CGR
CGR0
CGR1
CGRi
CGRn
GR
TO OAG
FGR
F
I
S
FGR0
FGR1
FGR I
FGRn
C3
C2
C1
C4
C5
30
ADDRESS COMPARE
FROM IIL
21
PC
DR
OP
20
SEP
10
23
22
IEP
11
25
24
FEP
12
TO OPF

FIG.6

FIG.7

FIG.8